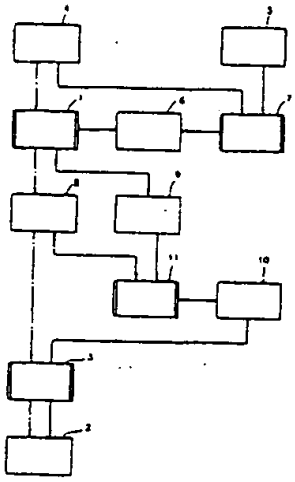
(54) PROGRAM LANGUAGE TRANSLATOR
(11) 3-85639 (A) (43) 10.4.1991 (19) JP
(21) Appl. No. 64-221573 (22) 30.8.1989
(71) FUJITSU LTD (72) HIROTOSHI YAMADA(4)
(51) Int. Cl⁵. G06F9/45

PURPOSE: To shorten time required for source analysis, and to improve the efficiency of translation processing by executing the source analysis only for a corrected part at the time of the translation of a source program after correction.

CONSTITUTION: After the correction of the source program, differential information 6 between the source program 4 before correction and the source program 5 after correction is delivered to a source analyzing means 1 by a hysteresis information control means 7. Then, a differential intermediate code 9 corresponding to the differential information 6 is substituted for a part corresponding to the corrected part of an intermediate code 8 before correction generated and preserved last time by the source analyzing means 1 by a substituting means 11. The substituted intermediate code 10 is delivered to a code generating means 3, and an object program 2 equivalent in a meaning to the source program 5 after correction is generated. Thus, when the source program is corrected and altered, since the source analysis is executed only for the corrected part, the source analysis need not be executed for the whole source program, and the processing time is shortened, and the processing efficiency is improved.



E5831

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　平3−85639

⑤Int. Cl.⁵　識別記号　庁内整理番号　⑬公開　平成3年(1991)4月10日

G 06 F 9/45

8724−5B　G 06 F 9/44　322 E

審査請求　未請求　請求項の数 1　(全10頁)

⑭発明の名称　プログラム言語翻訳機

⑳特　願　平1−221573

㉒出　願　平1(1989)8月30日

⑫発明者　山田　博敏　神奈川県川崎市中原区上小田中1015番地　富士通株式会社内

⑫発明者　池田　功　神奈川県川崎市中原区上小田中1015番地　富士通株式会社内

⑫発明者　中沢　通太　神奈川県川崎市中原区上小田中1015番地　富士通株式会社内

⑫発明者　山本　浩　福岡県福岡市博多区博多駅前1丁目5番1号　富士通九州通信システム株式会社内

⑪出願人　富士通株式会社　神奈川県川崎市中原区上小田中1015番地

⑭代理人　弁理士　松本　昂

最終頁に続く

明　細　書

1．発明の名称

プログラム言語翻訳機

2．特許請求の範囲

入力原始プログラムを解析し、命令単位の中間コードに展開するソース解析手段(1)と、

各中間コードに対応する命令列をオブジェクトプログラムとして生成するコード生成手段(3)と、

修正前原始プログラム(4)と修正後原始プログラム(5)との差分情報(6)を前記ソース解析手段(1)に渡す履歴情報管理手段(7)と、

前記ソース解析手段(1)により生成された修正前原始プログラム(4)に応じた修正前中間コード(8)の修正箇所に対応する部分を、前記ソース解析手段(1)により生成された差分情報(6)に応じた差分中間コード(9)に置換えて、前記コード生成手段(3)に渡す置換手段(11)とを具備してなることを特徴とするプログラム言語翻訳機。

3．発明の詳細な説明

概　要

プログラム言語で記載された原始プログラムを実行形式のプログラムに翻訳するプログラム言語翻訳機に関し、

翻訳処理時間が短く、処理効率の高いプログラム言語翻訳機の提供を目的とし、

入力原始プログラムを解析し、命令単位の中間コードに展開するソース解析手段と、各中間コードに対応する命令列をオブジェクトプログラムとして生成するコード生成手段と、修正前原始プログラムと修正後原始プログラムとの差分情報を前記ソース解析手段に渡す履歴情報管理手段と、前記ソース解析手段により生成された修正前原始プログラムに応じた修正前中間コードの修正箇所に対応する部分を、前記ソース解析手段により生成された差分情報に応じた差分中間コードに置換えて、前記コード生成手段に渡す置換手段とを具備して構成する。

産業上の利用分野

本発明はプログラム言語で記載された原始（ソース）プログラムを実行形式のプログラムに変換するプログラム言語翻訳機に関する。

プログラム言語（ソース言語）で記載された原始プログラムを、実在する計算機の上で実行させるためには、その目的計算機が実行できる形式のプログラム（オブジェクトプログラム）に、予め変換しておかなければならない。このために用いられるのがプログラム言語翻訳機であり、このプログラム言語翻訳機は、一般に原始プログラムを文法に従って解析する解析モジュールと、この解析データから原始プログラムと意味の等価なオブジェクトコードを生成するコード生成モジュール等からなる。これらのモジュールはさらにサブモジュールに分解することができ、前段のモジュールの処理結果を一旦、中間コードとしてファイルに格納し、これを次段の入力として処理する多重パス形式のものがある。

このようなプログラム言語翻訳機においては、

例えば数万行の原始プログラムに対して数行の修正を加えた場合に、修正を施したプログラム全体を再翻訳する必要があるため、時間的ロスが大きく、プログラム作成上の大きな障害となっている。このため、このような障害を除去し、効率的な翻訳処理を実現できるプログラム言語翻訳機の提供が要望されている。

従来の技術

第５図は従来のプログラム言語翻訳機の一例を示すブロック図である。４０はプログラマにより作成され、ファイルに格納されている原始プログラムであり、この原始プログラム４０は目的計算機で実行可能な形式のプログラムに変換するため翻訳機４１に入力される。翻訳機４１はソース解析部４２、最適化処理部４３、及びコード生成部４４から構成されている。ソース解析部４２はさらに字句解析部と構文解析部に分けることができ、字句解析部では入力された原始プログラム４０が名前、定数、特殊記号等のプログラムの基本構成

要素に分けられ、構文解析部ではこの基本構成要素からプログラムの構造を決定し、文法に合致するか否かが確かめられ、コード生成に必要な情報が取り出される。これらの処理を経た中間コード４５はファイルに格納される。

次いで、この中間コード４５は最適化処理部４３に入力される。最適化処理部４３では、実行時間の短縮化や記憶領域の縮小化等を目的とした最適化処理がなされ、この結果は中間コード４６としてファイルに格納されるとともに、コード生成部４４に入力される。

コード生成部４４では、入力された情報に基づいてプログラム中の変数や作業場所に対して実行時の記憶場所が決定され、構文解析の結果判明したプログラムの意味と、記憶割当の方針に従って、オブジェクトプログラム４７を生成し、問題が無ければオブジェクトプログラム４７をファイルに格納して処理を終了するようになっている。

プログラムにバグが発見される等してプログラムの修正が必要になった場合には、従来はプログ

ラムエディタにより原始プログラムを修正し、翻訳機４１により前回の処理と同様の処理を再度実施してオブジェクトプログラムを生成していた。

発明が解決しようとする課題

しかし、従来技術によると、例えば、原始プログラムを数時間を費やして翻訳処理してオブジェクトプログラムを生成した後、軽微なバグにより数行の修正を加えたような場合であっても、再度同じ処理を数時間かけて実施する必要があり、これは非常に効率が悪く、プログラム作成上の大きな障害となっていた。

本発明はこのような点に鑑みてなされたものであり、翻訳処理時間が短く、処理効率の高いプログラム言語翻訳機の提供を目的としている。

課題を解決するための手段

第１図は本発明の原理を説明するための図であり、本発明を構成する各手段とこれらの手段間でのデータの流れが示されている。同図中、一点鎖

線矢印は新規に作成された原始プログラムに対する翻訳処理を、実線矢印は修正が加えられた原始プログラムに対する翻訳処理を示している。

本発明のプログラム言語翻訳機は、入力された原始プログラムを解析し、命令単位の中間コードに展開するソース解析手段１と、中間コードに対応する命令列をオブジェクトプログラム２として生成するコード生成手段３と、修正前原始プログラム４と修正後原始プログラム５との差分情報６をソース解析手段１に渡す履歴情報管理手段７と、ソース解析手段１により生成された修正前原始プログラム４に応じた修正前中間コード８の修正箇所に対応する部分を、ソース解析手段１により生成された差分情報６に応じた差分中間コード９と置換えて修正後中間コード１０とし、これをコード生成手段３に渡す置換手段１１とから構成されている。

作　用

本発明によるプログラム言語翻訳機は、新規に

作成された原始プログラムに対して翻訳処理（１回目の翻訳処理）を行うときには、第１図中、一点鎖線矢印で示されているように、ソース解析手段１により原始プログラム４全体についての中間コード８を生成してこれを保存しておくとともに、コード生成手段３に渡してオブジェクトプログラム２を得る。

バグ対策等により原始プログラムを修正した場合（２回目以降の翻訳処理）においては、第１図中、実線矢印で示されているように、履歴情報管理手段７により、修正前原始プログラム４と修正後原始プログラム５の差分情報６がソース解析手段１に渡され、１回目（前回）ソース解析手段１により生成され保存された修正前中間コード８の修正部分に対応する部分を、この差分情報６に対応する差分中間コード９と置換手段１１により置換える。この置換えられた中間コード１０は、この時点で、修正後原始プログラム５全体をソース解析手段１で解析して生成された中間コードと同一のものとなっている。これをコード生成手段３

に渡して、コード生成手段３により修正後原始プログラム５と意味の等価なオブジェクトプログラム２が生成される。

このように、本プログラム言語翻訳機によれば、原始プログラムが修正変更された場合には、その修正された部分のみについてソース解析を行い、前回翻訳時に生成された修正前中間コードの修正箇所に対応する部分をこれと置換えるようにしているから、２回目以降の翻訳時には原始プログラム全体についてソース解析を実施する必要が無く、その処理時間を大幅に短縮することができる。

実　施　例

以下本発明の実施例を図面に基づいて詳述する。

第２図は本発明の一実施例を説明するためのブロック図であり、（Ａ）は新規に作成した原始プログラムに対しての翻訳処理を、（Ｂ）は１回以上の修正を加えた原始プログラムに対しての翻訳処理を示している。

２０はプログラムエディタを有する世代管理シ

ステムであり、プログラマはこの世代管理システム２０のプログラムエディタを用いて原始プログラム２１，２２を作成することができる。また、この世代管理システム２０は、バグ対策等のためにプログラムを修正した場合に、履歴情報管理ファイル２３に修正変更した情報等が自動的に記録されるようになっており、プログラムの履歴を管理する機能を有している。

本プログラム言語翻訳機は、この世代管理システム２０により作成された原始プログラム２１，２２を翻訳処理するものであり、新規に作成した原始プログラム２１と、これに１回以上の修正を加えた原始プログラム２２とで、その処理内容が異なっている。

まず、新規に作成された原始プログラムに対する処理を第２図（Ａ）を参照して説明する。世代管理システム２０のプログラムエディタにより、新規に作成され、ファイルに格納されている原始プログラム２１は目的計算機で実行可能なプログラムに変換するために翻訳機２４に入力される。

原始プログラム２１は、例えば第３図（Ａ）に示されるように記述されたプログラムである。尚、第３図（Ａ）中の行番号はこの言語の文法上記述されているものではない。

新規な原始プログラム２１に対しての本翻訳機の機能的な構成は第２図（Ａ）中、点線内に示されているとうりである。即ち、翻訳機２４はソース解析部２５、最適化処理部２６、及びコード生成部２７から構成されている。ソース解析部２４はさらに字句解析部と構文解析部に分けることができ、字句解析部では入力された原始プログラム２１が名前、定数、特殊記号等のプログラムの基本構成要素に分けられ、構文解析部ではこの基本構成要素からプログラムの構造を決定し、文法に合致するか否かが確かめられ、コード生成に必要な情報が取り出される。これらの処理を経た中間コード（ソース解析終了時の中間コード）２８はファイルに格納される。中間コード２８は例えば、第３図（Ｂ）に示されるような内容のものであり、中間コードはテキスト、行番号、辞書、及び修正

情報から構成されている。修正情報は後述する最適化処理において辞書に変更を生じた場合にその情報を記録する部分であり、この段階では使用されていない。

次いで、この中間コード２８は最適化処理部２６に入力される。この最適化処理部２５では、実行時間の短縮化や記憶領域の縮小化等を目的とした最適化処理がなされ、この結果は最適化処理終了時の中間コード２９としてファイルに格納され保存される。この中間コード２９は例えば、第３図（Ｃ）に示されるような内容のものである。同図（Ｂ）と比較して明らかなように、１００行目の辞書（Ａ＝Ｂ　Ｌ@）を削除し、替わって（Ａ／＝Ｂ　Ｌ@）を追加するとともに、３００行目のＬ１を削除することにより最適化が図られている。尚、ソース解析終了時の中間コード２８は最適化処理終了時の中間コード２９のファイル化と前後してファイル上から削除される。

次いで、この最適化処理終了時の中間コード２９は、コード生成部２７に渡される。コード生成

部２７では、中間コード２９の情報に基づいてプログラム中の変数や作業場所に対して実行時の記憶場所が決定され、構文解析の結果判明したプログラムの意味と、記憶割当の方針に従って、オブジェクトプログラム３０が生成され、このオブジェクトプログラム３０がファイルに格納されて翻訳処理が終了する。

次に、第２図（Ｂ）を参照して、２回目以降の翻訳処理について説明する。プログラムにバグが発見される等してプログラムの修正が必要になった場合には、世代管理システム２０のプログラムエディタにより前回作成した新規な原始プログラム２１の所定箇所を修正して原始プログラム２２を作成する。この原始プログラム２２は例えば、第４図（Ａ）に示されるように記述されたものであり、１００行目と５００行目に修正が加えられている。この修正を実施したときに、世代管理システム２０はこの修正内容を履歴情報として履歴情報管理ファイル２３に格納している。

そして、この修正後の原始プログラム２２の翻

訳処理を実施するときには、翻訳機２４には履歴情報管理ファイル２３に格納されている修正された内容（以下、差分情報）と、この原始プログラムが新規のものでないという情報が入力される。翻訳機２４ではこの原始プログラムが新規のものでないという情報を受けて以下のような処理を実施する。尚、各部の処理について第２図（Ａ）を参照して既に説明した部分についての説明は一部省略する。

即ち、差分情報はソース解析部２４で解析されて差分中間コード３１としてファイルに格納される。差分中間コード３１は例えば第４図（Ｂ）に示されるような内容である。次いで、この差分中間コード３１及び前回の翻訳処理において作成された中間コード（最適化処理がなされたもの）２９が中間言語再生成部３２に入力される。

中間言語再生成部３２では、前回の翻訳処理で作成された最適化終了時の中間コード２３（第３図（Ｃ））から、これに含まれている修正情報（最適化処理による修正情報）を基に最適化処理を実

施する前の中間コード２８（第３図（Ｂ））を復元する。そして、この復元された中間コード２８の修正箇所に対応する部分（１００行目と５００行目）を、差分中間コード３１（第４図（Ｂ））に置換えて、これを修正後中間コード３３としてファイルに格納する。この中間コード３３は例えば、第４図（Ｃ）に示されるような内容のものである。第３図（Ｂ）と比較して１００行目と５００行目が差分中間コード３１に置換えられていることがわかる。この時点で、修正後中間コード３３は修正された原始プログラム２２全体についてソース解析を実施した場合と同一の結果となっている。

次いで、修正後中間コード３３は最適化処理部２６に渡され、前回翻訳処理時と同様な最適化処理がなされて、例えば、第４図（Ｄ）に示されるような中間コードに変換された後、コード生成部２７に入力され、オブジェクトプログラム３０が生成されて翻訳処理が終了する。

２回以上の修正があった場合には、この２回目

の翻訳処理と同様の処理が繰り返し実行される。

上述したように、本実施例によれば、原始プログラムに修正が加えられた場合に、このプログラムに対する翻訳処理においては、原始プログラム全体に対してソース解析をすることがなく、修正された箇所のみについてソース解析を実施した後、前回翻訳時の中間コードの修正箇所に対応する部分をこれと置換えるようにしている。これにより、ソース解析に要する処理時間は修正箇所の大きさに依存することになり、プログラム全体をソース解析しないから、修正後原始プログラムの翻訳に要する処理時間が大幅に短縮される。

発明の効果

本発明は以上詳述したように、修正後原始プログラムの翻訳時には、修正箇所以外の部分についてソース解析を実施しないように構成したから、ソース解析に要する処理時間が大幅に短縮され、翻訳処理の効率を大幅に向上することができるという効果を奏する。

４．図面の簡単な説明

第１図は本発明の原理構成を示すブロック図、

第２図（Ａ）及び（Ｂ）は本発明の一実施例を示すブロック図であり、（Ａ）は新規に作成した原始プログラムに対しての翻訳処理を示す図、（Ｂ）は修正した原始プログラムに対しての翻訳処理を示す図である。

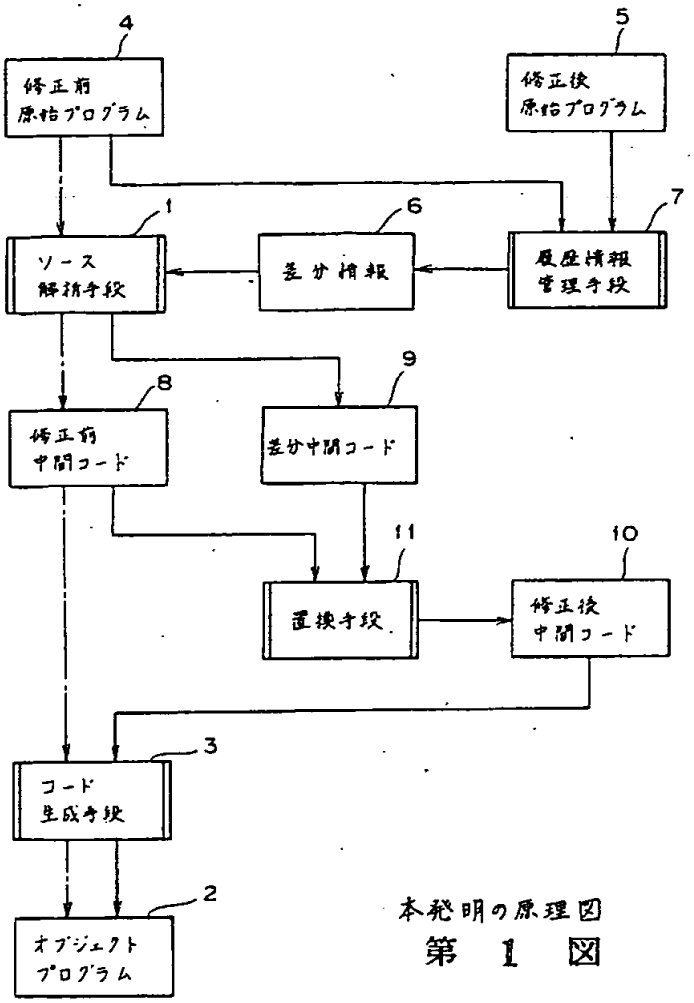
第３図（Ａ）～（Ｃ）は新規に作成した原始プログラム及び１回目の翻訳処理時に出力される中間コードの一例を示す図、

第４図（Ａ）～（Ｄ）は修正された原始プログラム及び２回目以降の翻訳処理時に出力される中間コードの一例を示す図、

第５図は従来技術を示すブロック図である。

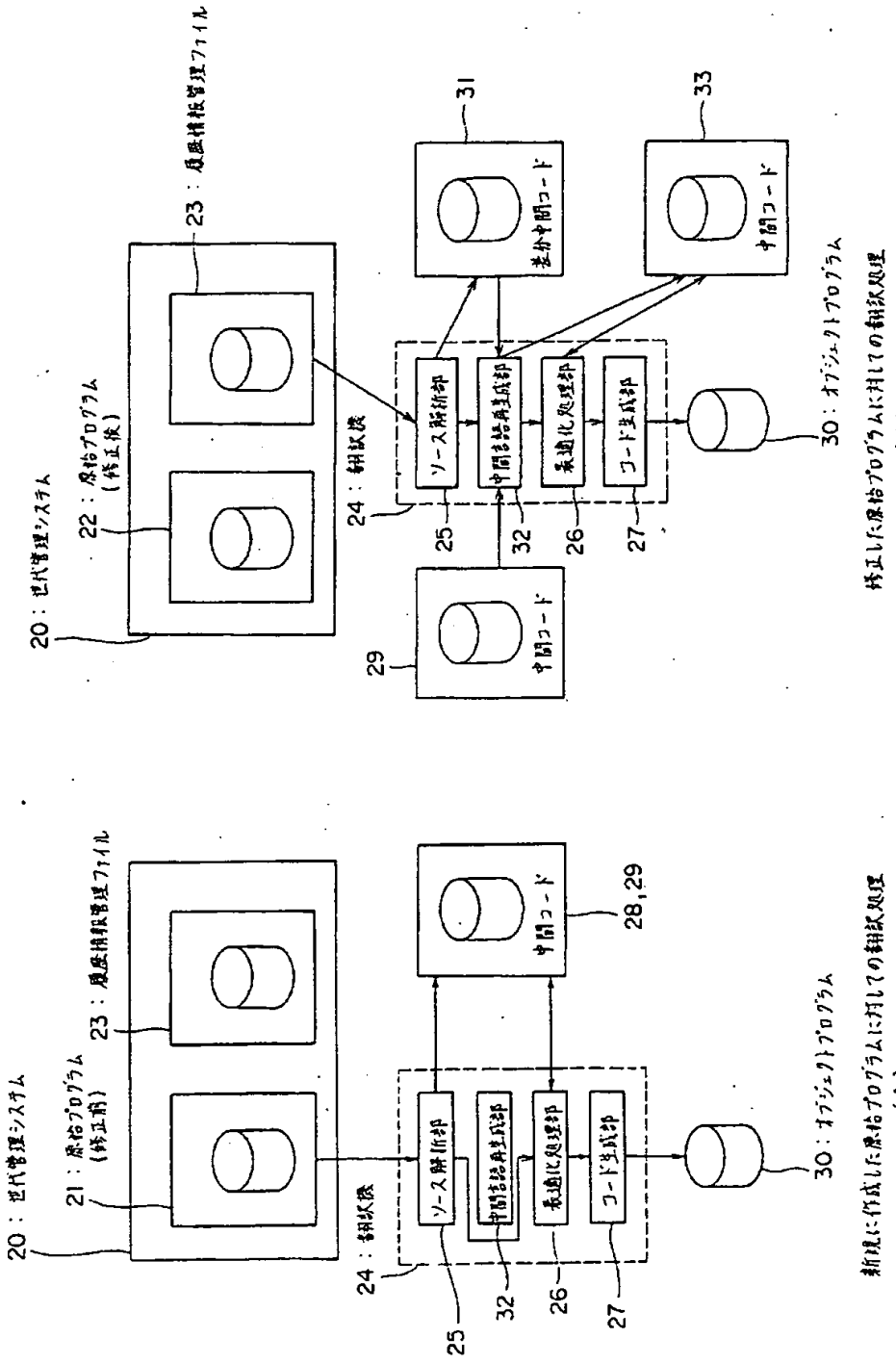
１…ソース解析手段、
２…オブジェクトプログラム、
３…コード生成手段、
４…修正前原始プログラム、
５…修正後原始プログラム、
６…差分情報、
７…履歴情報管理手段、
８…修正前中間コード、
９…差分中間コード、
１０…修正後中間コード、
１１…置換手段。

出願人：富士通株式会社
代理人：弁理士　松本　昂


4
修正前
原始プログラム
5
修正後
原始プログラム
1
ソース
解析手段
6
差分情報
7
履歴情報
管理手段
8
修正前
中間コード
9
差分中間コード
11
置換手段
10
修正後
中間コード
3
コード
生成手段
2
オブジェクト
プログラム

本発明の原理図

第 1 図



一実施例を示すブロック図

第 2 図

```
100     IF A = B
200       THEN
300         GOTO L1 ;
400     FI ;
500     C := A ;
600 L1: C := B ;
```

（A）入力ソース

テキスト	行番号	辞書	修正情報
STNO	100		
BNE		A=B L@	
STNO	200		
STNO	300		
B		L1	
STNO	400		
L@			
STNO	500		
LDST		C:=A	
STNO	600		
L1			
LDST		C:=B	

（B）ソース解析終了時の中間コード

テキスト	行番号	辞書	修正情報
STNO	100		
BEQ		A/=B L@	A
BNE		A=B L@	D
STNO	200		
STNO	300		
B		L1	D
STNO	400		
L@			
STNO	500		
LDST		C:=A	
STNO	600		
L1			
LDST		C:=B	

※ 修正情報 A⇒ 追加
D⇒ 削除

（C）最適化処理部終了時の中間コード

原始プログラム及び中間コードの一例を示す図（修正前）

第 3 図

```
100      IF A / = B ----⇨ 修正箇所
200        THEN
300          GOTO L1 ;
400      FI ;
500      C : = A + 1 ; ---⇨ 修正箇所
600 L1 : C : = B ;
```

（A）入力ソース

テキスト	行番号	辞書・・	修正情報
STNO	100		
BNE		A/=B L@	
STNO	500		
LDST		C:=A+1	

（B）ソース解析終了時の差分中間コード

テキスト	行番号	辞書・・	修正情報
STNO	100		
BNE		A/=B L@	
STNO	200		
STNO	300		
B		L1	
STNO	400		
L@			
STNO	500		
LDST		C:=A+1	
STNO	600		
L1			
LDST		C:=B	

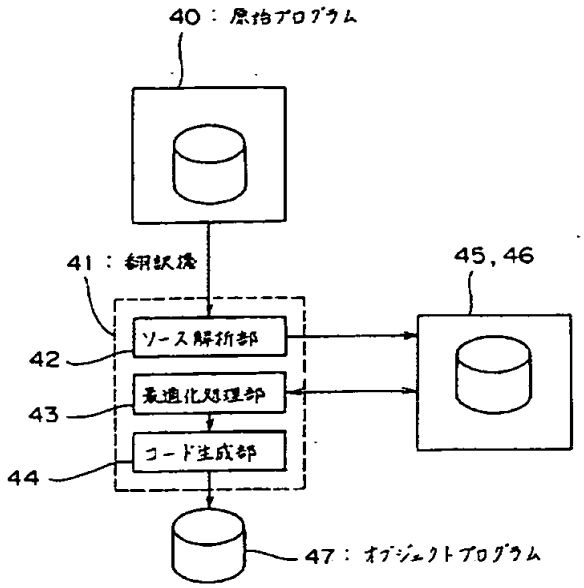
（C）中間コード再生成処理終了時の中間コード

テキスト	行番号	辞書・・	修正情報
STNO	100		
BEQ		A=B L@	A
BNE		A/=B L@	D
STNO	200		
STNO	300		
B		L1	D
STNO	400		
L@			
STNO	500		
LDST		C:=A+1	
STNO	600		
L1			
LDST		C:=B	

（D）最適化処理部終了時の中間コード

原始プログラム及び中間コードの一例を示す図（修正後）

第 4 図



従来技術のブロック図

第 5 図

第1頁の続き

⑫発 明 者　横 田　　学　福岡県福岡市博多区博多駅前1丁目5番1号　富士通九州通信システム株式会社内